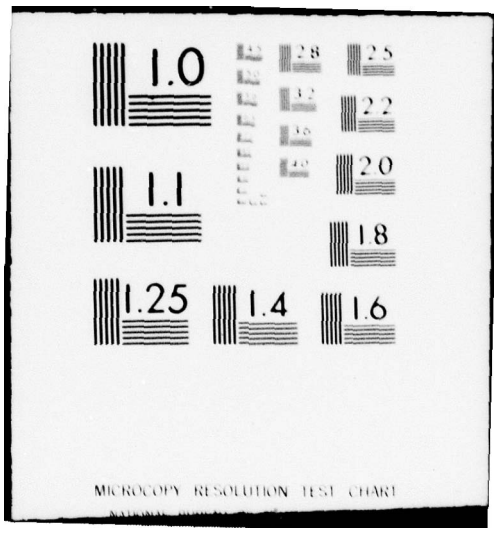
AD-A064 695 NAVY PERSONNEL RESEARCH AND DEVELOPMENT CENTER SAN D--ETC F/G 9/2
USER PERFORMANCE WITH A NATURAL LANGUAGE QUERY SYSTEM FOR COMMA--ETC(U)
JAN 79 R L HERSHMAN, R T KELLY, H G MILLER

UNCLASSIFIED NPRDC-TR-79-7 NL

1 OF 1
AD A064695

END
DATE FILMED
4-79
DDC


1.0
1.1
1.25
1.4
1.6
2.8
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART

NPRDC TR 79-7 January 1979

USER PERFORMANCE WITH A NATURAL LANGUAGE QUERY SYSTEM FOR COMMAND CONTROL

Ramon L. Hershman
Richard T. Kelly
Harold G. Miller

Reviewed by
Richard C. Sorenson

Approved by
James J. Regan
Technical Director

Navy Personnel Research and Development Center
San Diego, California 92152

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPRDC-TR-79-7
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): USER PERFORMANCE WITH A NATURAL LANGUAGE QUERY SYSTEM FOR COMMAND CONTROL
5. TYPE OF REPORT & PERIOD COVERED: Interim rept.
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Ramon L. Hershman, Richard T. Kelly, Harold G. Miller (Naval Ocean Systems Center)
8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Navy Personnel Research and Development Center, San Diego, California 92152 (Code 305)
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62763N; ZF55.521.019
11. CONTROLLING OFFICE NAME AND ADDRESS: Navy Personnel Research and Development Center, San Diego, California 92152 (Code 305)
12. REPORT DATE: January 1979
13. NUMBER OF PAGES: 53
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

12 54 p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

16 F55521 17 ZF5552101 9

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Natural language query systems
Interactive query languages
Man-computer communication
Data base access
Command and control

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Natural language query systems have been developed as potential aids to command control data retrieval processes involving large data bases. One such system, LADDER (for Language Access to Distributed Data with Error Recovery), was studied in order to identify significant performance characteristics associated with its use in a Navy command control environment. Ten officers received moderate training in LADDER and subsequently employed it in a search and rescue scenario. Both system and user performance were examined. Basic

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 03 19 014

390 772

patterns of usage were established, and troublesome syntactic expressions were identified. Design recommendations for the man-computer interface in command control query systems are discussed.

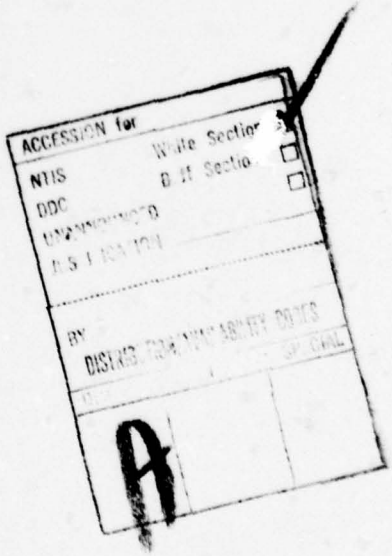
FOREWORD

This research and development was conducted in support of Exploratory Development Task Area ZF55.521.019 (Information Processing for Decision Making) in response to a request from the Advanced Command and Control Architectural Testbed (ACCAT) Project, Code 832, Naval Ocean Systems Center.

The effort supports the Navy's program to assess new and emerging technologies for their potential application to future command control systems. The data reported herein were collected in January 1978 and relate to the version of LADDER extant at that time. The results were made part of an ACCAT Project memorandum, "Performance of a natural language query system in a simulated command control environment," dated 19 May 1978 and prepared for the Naval Electronic Systems Command (ELEX 330).

The current address of Harold G. Miller is PME 108-3, Naval Electronic Systems Command.

DONALD F. PARKER
Commanding Officer



SUMMARY

Problem

There is a need to evaluate the performance of emerging information processing technologies for advanced command control applications. One such technology is LADDER (for Language Access to Distributed Data with Error Recovery), which is a prototype natural language query system for the retrieval of information from large command control data bases.

Objectives

The objectives of this effort were to obtain baseline quantitative data on the performance of LADDER and representative naval users in a simulated operational scenario, to identify problems that users encounter in interacting with LADDER, and to suggest remedial measures where appropriate.

Approach

Ten naval officers were given moderate training with LADDER and then served as operators in a search and rescue scenario. The users were required to formulate and enter queries in order to provide information to a hypothetical decision maker who would periodically make requests for information necessary to the conduct of the mission. An intelligent interface utilizing a Tektronix 4051 Desktop Computer was developed in order to provide training in LADDER, to supervise the management of the scenario, and to collect appropriate measures of performance. The evaluation was conducted at the Advanced Command and Control Architectural Testbed (ACCAT) Facility at the Naval Ocean Systems Center.

Findings

1. The users were able to retrieve an average of 91.6 percent of the 160 information items requested. The average number of queries made, however, was twice that which would be required by an expert LADDER user.

2. The system rejected 29.5 percent of the users' queries; 80 percent of these could be traced to errors of syntax. LADDER is faulted for often making excessively rigid syntactical demands. Certain query types (those involving time or distance and those permitting definitions by the user) were particularly prone to error.

3. The average time to initiate and complete a successful query was quite acceptable--103 seconds. The average component times were approximately 15 seconds for query formulation, 30 seconds for query entry, 15 seconds for parsing of the query, and 45 seconds for retrieval proper.

4. LADDER's average time to reject a query was excessive--38 seconds in addition to the time required for formulation and entry by the operator.

5. The users were generally favorable to the LADDER technology, although they cited substantial and specific difficulties in constructing acceptable queries.

Conclusions and Recommendations

1. LADDER exhibits rather impressive capabilities in interpreting natural language and retrieving information from a data base. But LADDER's natural language subset, at its current stage of development, is less than completely "natural" (rejection rate = 29.5%).

2. Selected types of queries are particularly prone to error because of LADDER's syntactical demands. Queries involving time or distance computations could be improved by expanding the permissible grammatical patterns. The queries that permit definitions by the user should be made more flexible and easier to use.

3. LADDER's rejection algorithm should be improved in order to reduce the excessive time required to determine that a query is faulty.

4. Fleet applications of natural language query systems must await the evolution and refinement of this protytype technology. Objective evaluations of system performance (in contrast to "demonstrations" and "subjective assessments") can best contribute to such evolution. LADDER's performance in other command control scenarios should be examined.

CONTENTS

LIST OF TABLES

FIGURES

INTRODUCTION

Problem

The retrieval of information from large command control data bases is a recurring problem in man-machine system design. Indeed, the advent of larger and faster computers has not guaranteed rapid and efficient access to military information. This is particularly true when a requirement exists to provide easy access for untrained users to a large, distributed data base.

Conventional systems thus force the user to preprocess and translate queries into an artificial, lower-level language, typically producing a stilted and formalized interaction. The burden in such interaction rests with specially trained operators, as in the Navy Worldwide Military Command Control System (WWMCCS) Query Module (Navy WWMCCS Software Standardization, 1975). The effect is to rule out a large class of more casual prospective users.

Natural language query systems comprise a newly emerging technology in information processing, and their potential applications to the Navy command and control environment need to be evaluated. Such evaluation should be done as early as possible in the technology's evolution.

Background

The Advanced Command and Control Architectural Testbed (ACCAT) was jointly established by the Defense Advanced Research Project Agency (DARPA) and the Navy in FY76 to address the following objectives:

1. Evaluate the operational utility and performance characteristics of emerging information processing technologies with respect to Navy command control requirements.

2. Determine functional specifications for growth and enhancement of command control capabilities.

3. Identify architectural alternatives based upon the successful development of information processing technologies.

4. Establish advanced methodologies and tools in an integrated testbed applicable to the continuing evaluation of new technologies within an operational command control context.

A site at the Naval Ocean Systems Center was designated as the principle ACCAT facility for the investigation of emerging technologies. The facility, in cooperation with other elements of ACCAT, serves as a predevelopment testbed that permits technologies and their applications to be examined in simulated operational environments, prior to commitment to costly development.

Query systems comprise one group of advanced technologies under investigation at the ACCAT facility. The system whose performance is reported here is called LADDER, for Language Access to Distributed Data with Error Recovery.

LADDER uses a subset of "natural language" and includes some advanced state-of-the-art techniques from the field of artificial intelligence as applied to a real-time performance system. It is the current product of an ongoing research project whose goal is to develop computer systems that can provide easy access for untrained users to large, distributed data bases. This development effort is under the sponsorship of DARPA and is being performed by SRI International, Menlo Park, California.

General Description of LADDER

Only an overview of the LADDER system is given in this section. For a detailed specification the reader should consult the report by Sacerdoti (1977), which forms the basis of the summary given here.

The LADDER system consists of three major functional components, INLAND, IDA, and FAM, that provide levels of buffering between the user and a data base management system (DBMS). LADDER employs the DBMS to retrieve specific field values from specific files just as a programmer might, so that the user of LADDER need not be aware of the names of specific fields, how they are formatted, how they are structured into files, or even where the files are physically located. Thus, the user can think he is retrieving information from a general information base rather than retrieving specific items of data from a highly formatted, traditional data base.

INLAND (for Informal Natural Language Access to Navy Data) translates the query from a restricted subset of natural language entered at the keyboard into a query or queries for specific fields of the data base. But INLAND makes no presumption about the way in which the information in the data base is organized into files. This query is then passed along to the second component of the system.

IDA (for Intelligent Data Access) then organizes the INLAND queries into a sequence of queries for various files within the data base. IDA uses a model of the data base structure to perform this operation and preserves the linkages among the retrieved records in order to return a readable answer to the user. In effect, IDA decides dynamically what logical view of the data base corresponds to the query, and decides dynamically how to satisfy the query in that logical view.

The third component, FAM (for File Access Manager) relies on a locally stored map showing where files are located throughout the distributed data base. When it receives a query expressed in the language of the DBMS, it searches its map for the primary location of the file (or files) to which it refers. It then establishes connections to the appropriate computer, logs in, opens the files, and transmits the query, amended to refer to the specific files that are being accessed.

The Data Base

The naval data base employed in this study is a relational data base named BLUEFILE that consists of 14 files (relations) and 73 fields (attributes). It contains information relationships of the following kind:

1. Ship's casualty and readiness status (condition of equipment and operating status).

2. Convoy information (departure, arrival, port destination, units, schedules).

3. Employment schedules of individual ships.

4. Ship's movement and track history (track history limited to current position and destination).

5. Individual ship and ship class characteristics (type and physical dimensions).

6. Characteristics of a particular unit (for example, CO's name and rank, operational control authority, fuel status, and homeport.

7. Weapon characteristics and classes of weapons on each ship.

The Situation Display

The Situation Display subsystem provides graphic output for selected requests from the LADDER query system. It contains map vectors in its local data base that are assembled to create geographical land mass outlines, centered and scaled on the display according to the map-drawing parameters specified by the user. This subsystem allows the generation and placement of symbols that designate ship positions and course on the display in the proper relationship to the map scale selected. For each displayed ship position, there is amplifying information placed in the map margin. This information includes the ship's name, class, unit identifying code, nationality, track history, and speed, as well as the date of the information. The Situation Display itself is a 25-inch color CRT.

Examples of LADDER Queries

Typical questions to LADDER might start with any of the following words:

1. what	6. when
2. where	7. is there
3. who	8. show
4. why	9. print
5. which	10. list

These first words are coupled into the remaining part of the question to extract information from the data base. Simple queries of the following type can then be formulated:

1. What is the name of the nearest ship to New York?
2. Who commands the KENNEDY?
3. How long would it take the INDEPENDENCE to reach 35-00n, 20-00w?
4. How many merchant ships are within 400 miles of the KENNEDY?
5. What is the readiness of the KITTYHAWK?

Compound questions can also be asked as long as their elements are properly related. Examples are:

1. What is the length, call sign, and ship class of the CONSTELLATION?
2. What is the course, speed, and destination of the STERETT?
3. List the U.S. Navy and merchant ships in the Med.

LADDER has an inferential feature to ensure that whenever any portions of the present query seem incomplete, LADDER will make the assumption that these relate to the previous query. Examples are:

1. What is the position of the JOUETT? [original query]

2. Her destination? [inferred query is "What is the destination of the JOUETT?"]

3. How long would it take the KNOX to reach the PECOS? [original query]

4. REEVES? [inferred query is "How long would it take REEVES to reach the PECOS?"]

5. What ships are within 1000 miles of Honolulu? [original query]

6. List their readiness, reason, and casreps. [inferred query is "List the readiness, reason, and casualty reports of those ships within 1000 miles of Honolulu."]

At the user's option, a one-word synonym may be assigned to an old word already in LADDER's dictionary. This can be used for brevity or when an alternate name is easier to use than the normally assigned one. Examples are:

1. Define golo to be like the ADMIRAL GOLOVKO.
2. Define frisco to be like San Francisco.
3. Define norlant to be like North Atlantic.

An additional feature of LADDER allows the user to substitute a new word (or string of symbols) for an old phrase already in LADDER's vocabulary. This is convenient when a relatively complex question is to be formulated a number of times. Examples are:

1. Define (where is TF.6)
   . . . like (where is the STERETT, HORNE, JOUETT).

After this definition has been made at the keyboard, each time the data base is queried about TF.6, the response will be for the three ships listed.

2. Define ($ KENNEDY)
   . . . like (what is the length and beam of the KENNEDY).

The symbol "$" has been substituted for the phrase "what is length and beam of the." Use of the symbol will shorten subsequent query time.

3. Define (* PENDLETON)
   . . . like (where is the PENDLETON).

The symbol "*" will now mean "where is the."

Objectives

The work reported here was designed to meet the following goals:

1. To obtain and document performance data associated with the use of the LADDER natural language query system in a Navy command control environment.

2. To establish, at the user interface, a flexible concept for evaluating LADDER (and other new command control technologies).

3. To develop a training package that introduces a naive user to LADDER's syntax and features.

4. To obtain baseline quantitative data on interactions between representative users and LADDER in a relevant operational scenario.

5. To identify problems at the user-LADDER interface and to suggest remedial measures or alternative designs.

6. To survey user opinion regarding the operation and potential usefulness of LADDER.

APPROACH

Overview

The overall approach was guided by the need to develop a proper environment for the LADDER evaluation. Such an environment must include, at a minimum, an operationally relevant scenario in which to exercise LADDER, a sample of representative Navy users to exercise the system, and proper tools for performance measurement, training of the users, and sufficient control of the user's task.

A search and rescue mission was selected as the scenario to be exercised. Such a mission is familiar to virtually all Navy users and calls for a variety of queries that address the major components of the data base. Thirteen naval officers, none of whom were familiar with natural language query systems, served as the user population.

An intermediate position was adopted with respect to the problem of training. LADDER is sufficiently demanding in its syntax and lexicon so that if no training were provided, the outcome of the evaluation could be predicted with certainty--namely, the technology would be severely limited as a tool for accessing a naval command control data base. On the other hand, there is little doubt that with extensive and specialized training in LADDER, Navy officers could indeed master the system. However, this would leave LADDER's response time as the only real datum of interest, and such information could be much more economically obtained in an off-line exercise of the query system. Accordingly, the users were given moderate training (approximately 1.5 hours) in LADDER syntax and vocabulary. The intention was to provide a fair test of the LADDER technology in a plausible setting and to make the evaluations as informative as possible.

Finally, it was essential to provide unobtrusive tools for the objective measurement of both the user's and LADDER's performance. Substantial effort was devoted to the development of such tools, without which any evaluation would have been of limited value.

Users

A variety of roles are possible for users of a query system like LADDER. At one extreme, specially trained personnel are the only users who would interact with the query system. In this role the user is essentially a typist, entering queries verbatim that have been formulated by others. The focus in this scheme would be on LADDER's performance to the virtual exclusion of the user, and indeed, this "typist role" is best investigated in an off-line, noninteractive setting.

At the other end of this continuum, the user is a decision maker who is responsible for directing a complex naval operation. Such a user must analyze the entire situation and determine what information is needed. The user would then have to formulate LADDER queries to satisfy his information needs, enter these queries, analyze LADDER's responses, and continue to generate additional queries. This situation would provide a potentially rich mix of user-LADDER interactions, but it would confound the performance of the decision maker with

that of LADDER. That is, one could not separate the influences of the decision maker's problem solving skill from his proficiency with LADDER on the basis of scenario performance. As a result, very little could be determined regarding LADDER's utility in command control operations. Moreover, the results would be largely anecdotal, since each user would have complete control of the number, type, and relevance of the queries asked. Although a study of LADDER's usefulness to a decision maker is certainly germane in the design of command control systems, it is somewhat premature at this point.

A compromise between the typist role and the decision maker role was selected for this evaluation. In particular, the user served as an "operator" and was required to formulate and enter queries that were to provide information for a third party. This hypothetical third party was the decision maker, who periodically made broad, compound requests for information. The user responded to these information requests and was responsible for composing and entering the specific LADDER queries necessary to provide the requested data. In this role, the user's dialogue with LADDER was completely under his control, although the content of his queries was dictated by the external requests from the surrogate decision maker. Thus, an acceptable degree of realism was provided without unduly sacrificing control of the task. Moreover, the users' performance was less dependent on individual styles of problem solving or decision making. At the same time, significant cognitive demands were imposed on the user by the requirement to construct queries that answer specific information requests.

Thirteen naval officers who were serving in research and development positions volunteered to participate as users. The data gathered from three of these were incomplete and were excluded from subsequent analysis. None of the remaining ten officers had previous experience with LADDER or with any other natural language query system. However, eight of the participants knew at least one programming language, and six knew two or more languages. Seven of the users were Lieutenant Commanders, and three were Commanders. Their median level of command control experience was 6 years and ranged from 0 to 17 years. Five of the users reported that they had prior operational experience with an actual search and rescue operation.

Instrumentation

In LADDER's normal mode, a standard CRT terminal serves as the interface between the user and the host computer. That is, the terminal is entirely passive as it transmits and displays all communications between the user and host. This configuration was ill-suited to the present evaluation for the following reasons: The user and LADDER comprise a "closed" system that is isolated from manipulation and measurement by the researcher; LADDER provides no training for the user; LADDER communications are system-oriented rather than user-oriented; record-keeping by LADDER is incomplete and inaccurate; and the editor in LADDER is limited and awkward to use.

These shortcomings all stem from the fact that LADDER was not designed as an evaluation tool or indeed for the ultimate Navy user, but rather as a prototype technology *per se*. Short of revising the LADDER software, which was beyond the scope of the present effort, a critical step was to modify the existing interface by opening it to intelligent and flexible intervention.

The Tektronix 4051 desktop computer was selected to provide the modified interface. This unit is fully programmable in a modified version of BASIC and provides both alphanumeric and graphic display. Storage and retrieval of programs and data are achieved with an associated magnetic tape cartridge. In addition, the device is readily connected to other computers by its data communications interface and with miscellaneous peripheral devices by its general-purpose interface bus. Thus, by virtue of its software and communication facilities, the Tektronix 4051 permits enhanced data collection and storage, the filtering of LADDER output, an improved editing system, and overall management of the LADDER evaluation--all without disturbing normal system operations.

Figure 1 shows the tasks performed by the interface and its relation to the user and to the host computer. Thus, the Tektronix data communications interface linked the 4051 to a DEC PDP-10 in which the LADDER system resided. Communication over this link was set at 300 baud, since characters were occasionally lost at higher transmission rates. A timing generator (Hewlett-Packard model 59308A) was connected to the 4051 via the Tektronix general-purpose interface bus. The 4051 operated the timing generator under program control in order to measure the latencies of various user and LADDER responses.

Procedure

Orientation

Several days prior to his participation, each officer received an information packet that outlined the nature of the project and emphasized the importance of each user's participation. In addition, the LADDER query system was briefly described along with the BLUEFILE data base. Finally, the search and rescue scenario was described and the standard procedures for its execution were discussed.

Each user participated individually in a single experimental session that lasted between 3 and 6 hours, depending on the user's success with LADDER. Upon arriving at the ACCAT facility, each officer was given a brief overview of the layout and purposes of ACCAT. After a brief introduction to the Tektronix 4051 terminal and the Situation Display, the training program proper was initiated.

Training

In order to introduce the user to LADDER's features and syntax, a training package was developed to operate on the Tektronix 4051 terminal. The package was designed to allow a user to progress at his own rate through a series of activities intended to familiarize him with LADDER and with the terminal itself. The entire training session, which typically lasted approximately 90 minutes, consisted of three parts. First, the user proceeded through a tutorial, which discussed LADDER's grammar and the techniques for query entry and editing. Then, the user practiced entering and editing queries on the 4051 terminal. In the final training segment, the user practiced the actual composition of LADDER queries. Throughout the training session, an effort was made to maintain a general, user-oriented description of LADDER's characteristics. Explicit, detailed discussions of LADDER's idiosyncrasies were avoided whenever possible.

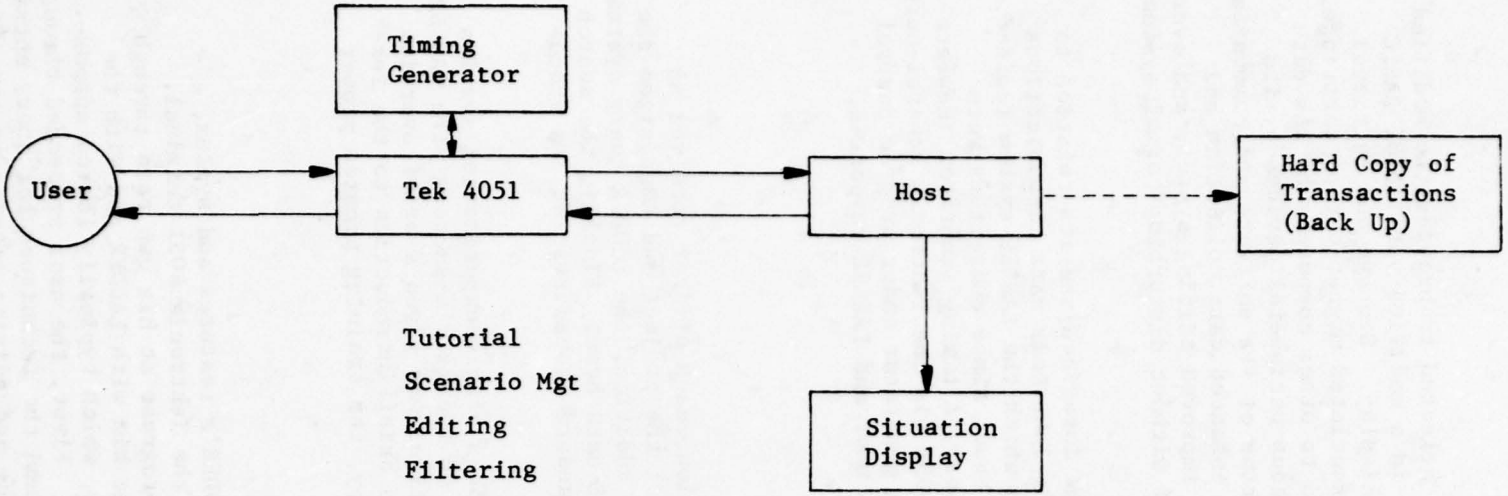


Figure 1. Tasks performed by the intelligent interface and its relation to the user and host computer.

LADDER Tutorial. The tutorial was designed to rapidly acquaint a naive user with LADDER's capabilities and procedural requirements. The text and graphics used in this tutorial were displayed on the screen of the Tektronix 4051. Users were free to progress at their own pace; typically, the tutorial was completed within 30 minutes.

The first part of the tutorial outlined again the purpose of the project and noted the natural language structure of LADDER. The user was then shown a representative sample of LADDER's vocabulary, which included descriptors, names of equipment, personnel, and ships. The sample served to emphasize the natural language aspects of LADDER and to suggest the types of information residing in the data base.

Following this overview, a lengthy discussion of LADDER's syntax was presented. This discussion explained the structure of each of the major types of LADDER queries. Numerous examples were provided to illustrate the use of general information queries, of compound queries, and of queries that referred back to earlier questions. The use of the Situation Display and of both types of DEFINE commands was also described. Hard copies of this portion of the tutorial were made available to each user as a LADDER Reference Folder, so that LADDER's query formats could be consulted, as required, during the subsequent scenario.

The final portion of the tutorial instructed the user in the procedures for query editing. Although the modified user-LADDER interface afforded the opportunity to employ a large number of editing features, only those available in the standard version of LADDER were used. Thus, users could delete individual characters (by the RUBOUT key) or delete an entire query (by a Function key) and start over. The use of these editing functions was explained and demonstrated.

Query Entry Practice. Although the tutorial provided an overview of LADDER and its utilization, the user had no chance to operate the terminal's keyboard or to practice the editing procedures. Therefore, query entry practice was provided at this point. This practice session also permitted each user's typing proficiency to be monitored, so that baseline rates of typing speed and accuracy could be established.

Here the user was given preformulated queries and was instructed to enter them verbatim. Although LADDER is able to correct minor spelling errors, the user was asked to enter the queries without error, and to use the editing functions if necessary. The Tektronix 4051 supervised this practice session by sequentially displaying the queries and providing immediate feedback to the user concerning his typing accuracy. For each query, the total entry time and the accuracy of the entry were stored on a data tape for later analysis. Each user was required to enter five practice queries, although additional practice was made available if desired. Note that a collateral function of this session was to expose the user to additional queries that were acceptable to LADDER. This added exposure was expected to help broaden the users' awareness of LADDER's capabilities.

Query Composition Practice. The user was next given the opportunity to actually compose LADDER queries. In this way, the grammar of the tutorial could be exercised, and major misunderstandings could be corrected. Ideally,

the user would receive feedback from LADDER regarding the acceptability of each practice query. To conserve time, however, direct interaction with LADDER was bypassed, and a self-paced paper-and-pencil procedure was substituted. Using this procedure, the user progressed through a booklet that contained a series of information requests posed as directives from an external decision maker. That is, specific information was requested by natural language commands that were not necessarily acceptable to LADDER. Examples of such requests are "Find out what kind of cargo is on the MCGRAW," "Locate the nearest aircraft carrier to 37-40N, 174-00W," "Identify the KENNEDY, LEAHY, and DALE as a task group." The user's task was to compose acceptable LADDER queries that would satisfy each request.

After writing a given query in the booklet, the user was given verbal feedback regarding its acceptability to LADDER. This feedback was provided by a highly experienced LADDER user and was purposely kept general. For unacceptable queries, only the first error (reading from left to right) was identified. Detailed discussions of LADDER's syntax and vocabulary were avoided, although any specific questions from the user were answered. Following the feedback, the user turned to the next page in the booklet, which contained several queries that would satisfy the information request and would be accepted by LADDER. By studying these examples, the user could discover other ways to construct LADDER queries and also could deduce reasons why the composed query may have been unacceptable.

The information requests were specially chosen to sequentially address each of the major types of LADDER queries. These included queries about basic status information, single queries that requested multiple data, queries that referred to previous questions, queries involving time or distance calculations, queries that addressed the Situation Display, queries used to define a new word, and queries to define a new phrase. At least five requests of each type were presented. An effort was made to ensure that each user had at least one acceptable query of each type prior to completing the practice session.

The Search and Rescue Scenario

In order to determine LADDER's potential utility in the command control environment, it was necessary to examine user behavior within a representative scenario. This would not only encourage the users to exploit the data base in a realistic progression but would also place an appropriate demand on LADDER's capability to interpret language natural to Navy command control operations.

A search and rescue mission was selected since this type of operation calls for a progression of activities that exercise a wide range of status and distance data, the details of the mission are familiar to most Navy officers, and the sequence of activities is unambiguous and presents the opportunity for generating many queries while logically remaining within the simulated environment.

A scenario was written in which an American tanker (the PECOS) was reported to be on fire in an area approximately 900 miles northwest of Hawaii. A watch officer at the Joint Rescue Command Center in Honolulu was considered to be in control of the search and rescue operation, and periodically he would

request information from the LADDER user/operator. In fact, the requests were composed ahead of time and presented on the Tektronix 4051 display as the scenario unfolded.

Several modifications to BLUEFILE, LADDER's unclassified data base, were made in order to accommodate the scenario. In particular, the positions and characteristics of several ships in addition to the PECOS were added or changed in BLUEFILE.

Scenario Management and the Flow of Events

As discussed above, the user's role was that of a LADDER operator who was to compose and enter queries that would satisfy the information requests of an external decision maker (here the watch officer in Honolulu). Complete management of the scenario was achieved by software especially written for the Tektronix 4051. From the user's viewpoint, however, except for the intrusion of the watch officer, the process very much resembled a straightforward dialogue with LADDER. The flow of events appears in Figure 2.

Information Requests. After initialization of the system, the first of the requests was displayed to the user. Subsequent requests were presented one at a time to allow a progression through the simulated search and rescue mission. There were 15 requests in all, and these appear in Appendix A. It should be noted that the requests were only marginally compatible with LADDER's grammar. In general, reformulation by the user was necessary to render them acceptable. Also note that an efficient mission could be conducted with fewer requests than these. However, the efficient conduct of the scenario was deemed secondary to the additional opportunities for composition and entry of LADDER queries.

User Entry and Editing. The user composed a query, then entered and edited it via the Tektronix 4051 keyboard.

Timing of User Processes. The time for query formulation was taken as the interval between the presentation of the request and the entry of the first character by the user. The time for query entry was taken as the interval between the entry of the first and last characters. Times were obtained under program control by the Hewlett-Packard timing generator.

Use of LADDER Reference Folders. After a given query had been entered, the interface program asked the user if the grammar reference folder had been used to construct or verify the query. The response was made using the keyboard.

Sending the Query to LADDER. The user's query was transmitted to LADDER over the data communications interface connected to the PDP-10.

Polling and Display During the Parse-or-Rejection Cycle. The program next waited for a response from LADDER to determine if the query had been parsed or rejected. All intermediate messages were displayed, so that LADDER's attempts to correct spelling errors and to clarify ambiguous queries would be apparent to the user.

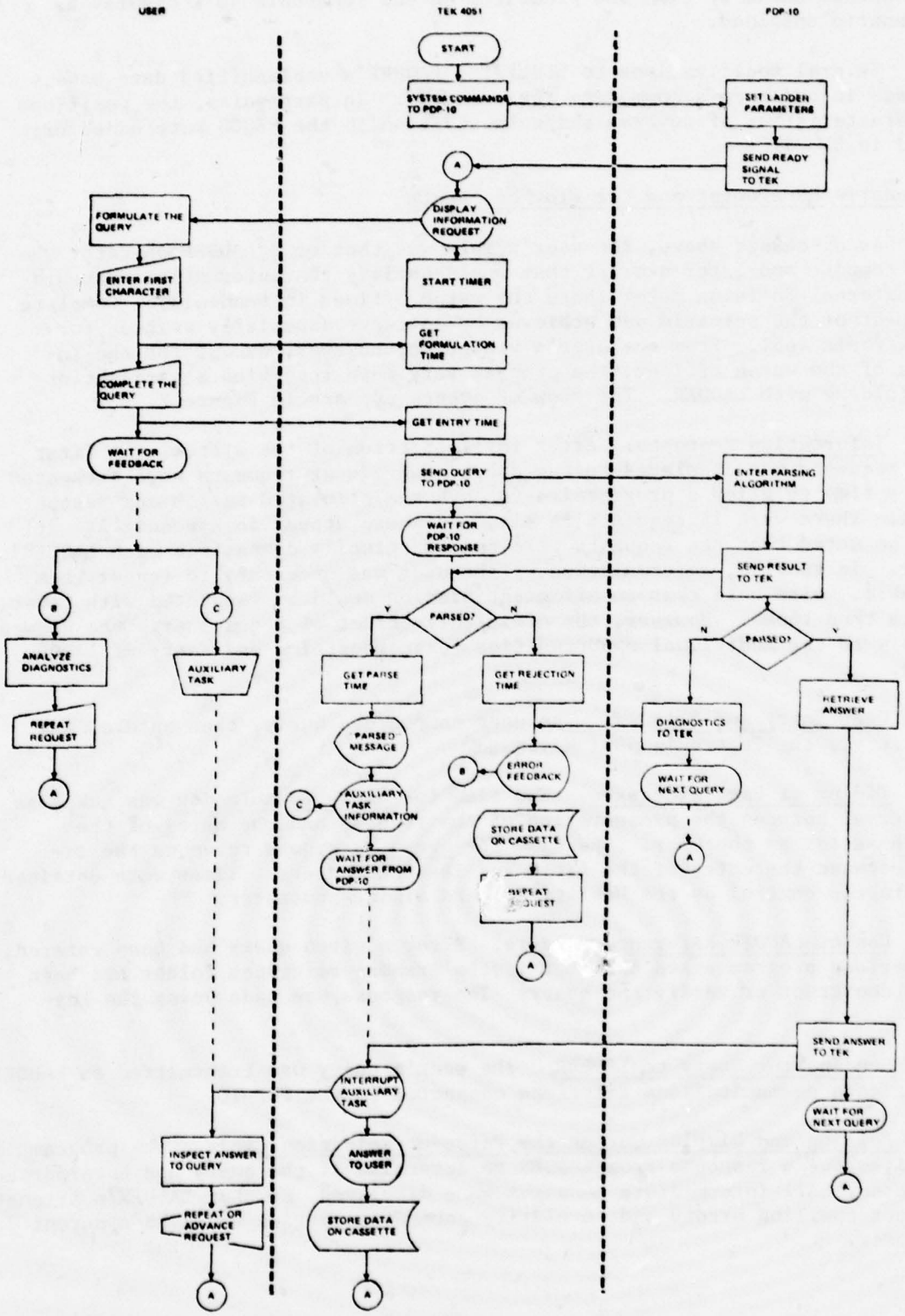


Figure 2. Flow of events for supervision of the user-LADDER interactions.

User Feedback and Timing of the Parse-or-Rejection Cycle. If LADDER rejected the query, the program measured LADDER's rejection time. It then informed the user of the failure to parse and also displayed any diagnostic information provided by LADDER. The unfilled request from the "watch officer" was again displayed; reformulation and entry of the query proceeded as before.

If LADDER parsed the query, the time-to-parse was measured, and the display informed the user that the query was acceptable. LADDER meanwhile proceeded to access the data base and to retrieve the desired information.

Auxiliary Tasks for the User. LADDER's retrieval time (including data base access) may exceed 60 seconds. Rather than have the users unoccupied during this period, several auxiliary tasks were introduced. Display messages from the "watch officer" initiated these user-paced activities, which included the interpretation of weather charts and the processing of aircraft readiness information.

Polling the Host During the Retrieval Cycle. For parsed messages, the program again polled the host and waited for LADDER's answer to the query. (The user at such time was engaged with the auxiliary task.) Extraneous messages from LADDER were ignored.

User Feedback and Timing of the Retrieval Cycle. Once the requested data had been retrieved, the program measured LADDER's retrieval time and sounded a bell at the 4051 console to alert the user. LADDER's answer was then displayed with a custom format that included the original request and the user's actual query.

Sequencing of the Information Requests. After receiving LADDER's answer, the user indicated whether he wished to proceed to the next request or to repeat the present one. This option provided substantial freedom in query composition. For instance, the user might include all of the necessary data in a single query, fragment the request into multiple queries, or define new words or phrases at any time.

Data Storage. The user's queries, LADDER's responses, the user's times for query entry and formulation, and LADDER's times for rejection or parsing and retrieval were stored on the 4051's tape cartridge for later analysis. (Since heavy loading on the PDP-10 system is known to inflate LADDER's processing times, the system load level was kept at or below 1.75 jobs in queue whenever possible.) Finally, some accounting data and minor behavioral items were stored, namely, the usage (if any) of the grammar reference folder and the number of times that the given query was "aborted" and restarted.

Terminating the Scenario. The scenario continued until the last request had been answered or until approximately 4 hours had elapsed. A typical scenario session lasted approximately 3 hours.

Debriefing

At the conclusion of the scenario, each user completed a questionnaire that sampled various opinions about LADDER's use. A copy of the questionnaire appears in Appendix B; the format consisted largely of five-point scale items.

These items surveyed user opinion about the adequacy of the training session, the desirability and ease of use of the various LADDER features, the interpretability of LADDER's output, and the potential utility of LADDER in an operational command control situation. In addition to these items, the users were encouraged to discuss any desirable additional features, significant limitations, or potential usage patterns of LADDER.

RESULTS AND DISCUSSION

Two of the 13 users experienced great difficulty with LADDER and were able to make little or no headway with the scenario. Their data were discarded from the analysis. The data from another user were lost due to an equipment malfunction, leaving a sample of ten users.

Pre-scenario Proficiency

An indication of each user's pre-scenario proficiency with LADDER was obtained from the query composition practice session. Here each user composed 39 total queries involving all of the major LADDER constructions. Overall, 80.5 percent of the 390 practice queries would have been acceptable to LADDER. However, three of the constructions posed special problems. These were the Time/Distance queries, the Situation Display commands, and the Define Phrase commands, each of which had an error rate of 34 percent. Difficulties with these constructions are attributed to their more demanding syntactic requirements.

Keyboard entry skill was also measured before the start of the scenario. The mean inter-keystroke time was 0.51 seconds, with a standard deviation of 0.12 seconds; 77 percent of the queries were entered accurately. While they were by no means skilled typists, the users did evidence substantial familiarity with the keyboard. A typical query was input in approximately 20 seconds.

Efficiency of Query Composition

Although the requests from the "watch officer" were quite specific, LADDER's flexibility permitted the users to employ a variety of approaches in acquiring the required information. Thus, there was no guarantee that all of the requests would actually be fulfilled. Also, the volume and information density of the users' queries certainly relate to the issue of efficiency.

Fulfilling the Information Requests

The 15 information requests called for a total of 160 information items to be retrieved and, for the most part, the users fulfilled the requests in a systematic fashion. In particular, the users retrieved an average of 91.6 percent, or 146.5 items (standard deviation = 12.3). Four of the ten users obtained 100 percent of the requested items, and all but three users retrieved more than 90 percent of the required information. Departures from the scenario's information requests were rare. Only four of 366 total queries (1.1%) addressed information that was clearly extraneous to the requests. As these data indicate, the users were able to formulate mission-related queries and to obtain the stipulated information from LADDER.

Query Volume and Information Density

The number of queries made by each user appears in column 2 of Table 1. Thus, the ten users submitted a total of 366 queries to LADDER (mean = 36.6 queries, standard deviation = 7.9), with the individual volume ranging from 28 to 57. Study of the 15 requests revealed that an "expert" LADDER operator could have retrieved the desired information in 18 queries. The average user,

then, made twice the queries necessary to access the requested data. The mean number of information items retrieved per query was 4.0 for the users versus 8.9 for the presumed expert. It follows that query efficiency was 45 to 50 percent for the subject users.

Table 1

Query Volume, Query Rejection, and the Sources of Query Rejection

User #	Number of Queries	Number Rejected	% Rejected	Number of Rejections		
				Syntax	Typing	Vocabulary
1	38	20	52.6	15	3	2
2	28	9	32.1	7	2	0
3	29	5	17.1	2	2	1
4	32	4	12.5	3	1	0
5	34	6	17.6	5	1	0
6	57	8	12.3	4	0	4
7	32	9	28.1	8	0	1
8	40	19	47.5	17	2	0
9	39	14	35.9	11	3	0
10	37	14	37.8	14	0	0
Mean	36.6	10.8	29.4	8.6	1.4	0.8
Standard Deviation	7.9	5.7	14.4	5.4	1.2	1.3

Query Rejection and Its Sources

Efficient query composition relies heavily on the user's ability to construct queries that are acceptable to LADDER and of high information density. Clearly, if a query is not acceptable, then it must be reworded and subsequently reentered. Similarly, if a user submits several simple queries to LADDER instead of combining these into a single compound query, further inefficiency is introduced.

Although both sources of inefficiency were observed, query rejection was a substantially greater problem. The number and percentage of queries rejected by LADDER are shown in columns 3 and 4 of Table 1. Overall, 108 (29.5%) of the 366 queries were rejected, with the individual rejection rate ranging from 12.3 percent to 52.6 percent. Such a high level of query rejection indicates that significant deficiencies exist in the user-LADDER system.

Each query rejection was analyzed and attributed to one of three sources: syntax errors that violated LADDER's rules of grammar, typing errors that LADDER could not correct, or vocabulary errors that occurred when the user employed a term that was not in LADDER's lexicon. Figure 3 summarizes these data relating to query rejection. Thus, of the 108 queries that were rejected, 86 (79.6%) were due to syntactical problems, 14 (13.0%) could be attributed to typing errors, and 8 (7.4%) were due to vocabulary errors. The data for individual users appear in columns 5, 6, and 7 of Table 1.

Syntax Errors

Clearly, syntax errors were the major cause of the systems' high rejection rate. One may fault the user for his failure to compose in acceptable syntax, the training regimen for failure to prepare him properly, or LADDER for making excessively rigid syntactical demands. It appears that the user should be absolved. After all, he is an intelligent "natural language expert" and a professional in the naval subject matter at hand.

The training regimen was substantial (1.5 hours) if not exhaustive. But most importantly, it did not focus explicitly on LADDER's syntactical idiosyncrasies. There is little question that with more intense and specialized training, the users could have produced a much higher rate of acceptable LADDER queries. But such a tactic would have been contrary to the fundamental objective, which was to evaluate the LADDER technology "as is" and with "fair rules of the game" prevailing--that is, in a plausible setting with plausible users given moderate training. To have provided more specialized training would have biased the evaluation in favor of LADDER and would not have fairly revealed the system's shortcomings. Moreover, such detailed training should not be necessary for a natural language system.

It appears, then, that LADDER, with its often rigid syntactic rules, is at fault, despite its rather impressive capabilities in interpreting natural language. In this light, the users' queries are perhaps best viewed as a high-level programming language. In effect, the user is a programmer and becomes subject to the rules of the programming language. Thus, whenever LADDER fails to accept familiar natural language constructions or demands unnatural syntax, the result is a programming error, and the user's query is rejected. As we have seen, 79.6 percent of the 108 rejected queries could be traced to syntactic difficulties.

Typing Errors

Considering that each query was not only checked and edited by the user but also submitted to LADDER's parser, this error rate (13.0%) was surprisingly high. Nevertheless, based on the pre-scenario typing practice, a 23 percent error rate would be anticipated if LADDER's error-correction aiding were not provided.

Vocabulary Errors

As for errors in vocabulary, it quickly became apparent that LADDER's lexicon does not include such familiar terms as "range," "nm" (nautical miles), and "dst" (destination). To be sure, these terms could be defined by the user, but this should not be necessary with a system that is presumably designed for Navy use. (Note that the effect is to immediately tell the user that the system does not understand his language.)

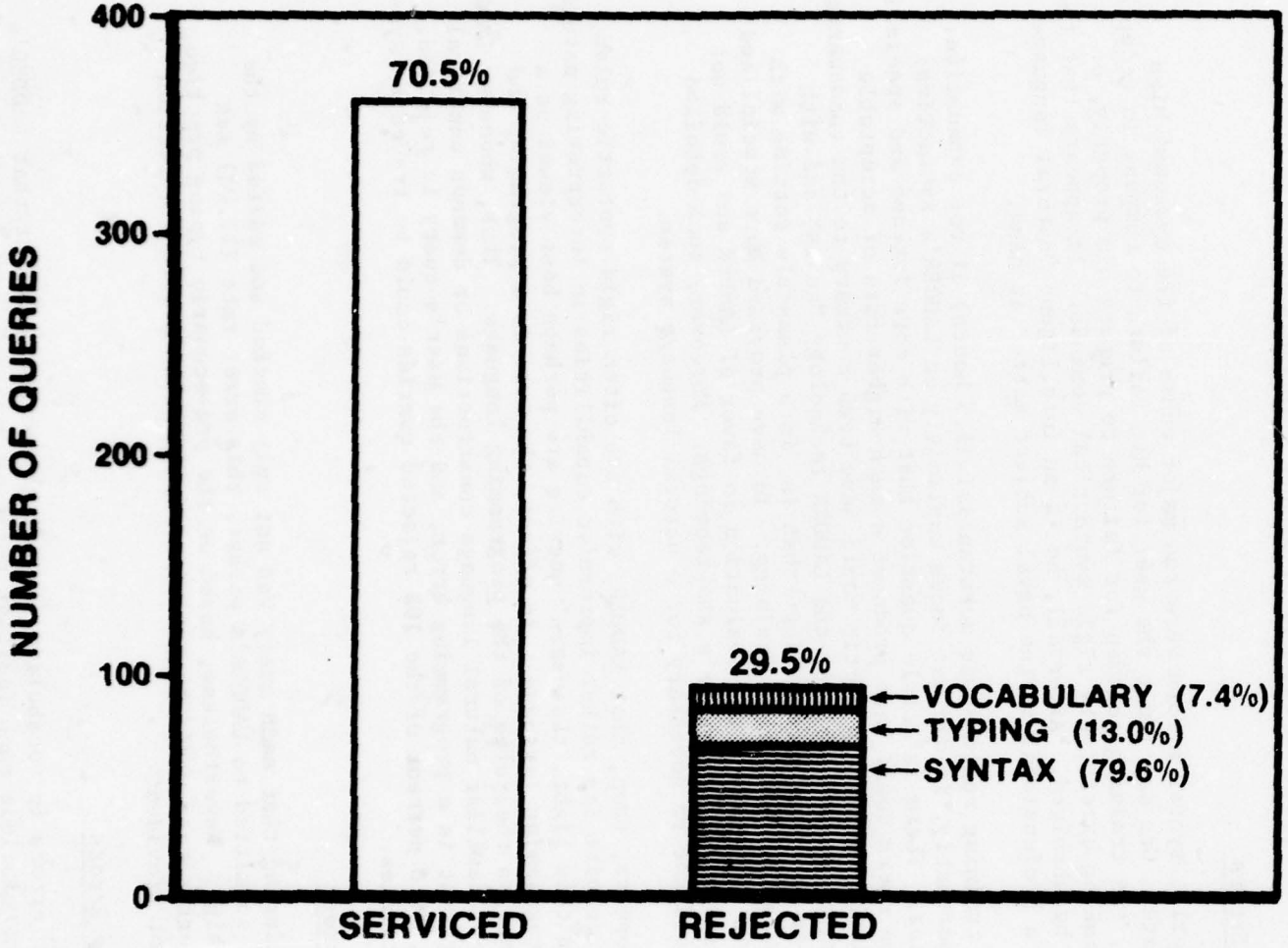


Figure 3. Serviced queries, rejected queries, and the sources of query rejection.

Another source of confusion involved the use of the term "opcon." Many users insisted on asking "Who has the opcon of . . .?" However, LADDER rejects queries concerning "opcon" that begin with "who." The users' protocols suggest that "who" is more natural for many Navy users.

Analysis of Syntactic Errors

As one might suspect, all queries are not equally likely to be rejected. Rather, certain types of user requests are particularly prone to rejection by LADDER's parsing algorithm, while others are much more likely to be acceptable. A suitable partitioning of the query set should serve to focus attention on specific problems that might be remedied in future evolutions of LADDER.

Types of Queries

While no ready-made partition of the possible LADDER queries exists, certain types of queries do cluster together and suggest membership in a common class. For example, the calls "Select a map," "Erase," and "Show" all address the Situation Display and thereby form a natural class. Such natural clustering, together with inspection of the users' data, yielded the six query types described below. These categories were adopted instead of those used in the training session in order to minimize overlap and to reveal special vulnerabilities to error.

Explicit--General Information.[1] All referents are named, but queries involving time or distance are specifically excluded. The following are acceptable examples:

What is the position of the KENNEDY?
What is the nationality and owner of PECOS?
Name the commanding officers of RATHBURNE and KNOX.

Explicit--Time/Distance. All referents are named, and the query addresses at least one characteristic involving time or distance. Such queries are particularly vulnerable to errors in syntax and, for this reason, are identified as a separate type. The following are acceptable examples:

What is the distance between KENNEDY and Hong Kong?
How long will it take RATHBURNE to reach PECOS?
How far is PECOS from Honolulu?

---

[1]To begin the scheme, "explicit" queries were first distinguished from "implicit" ones. The former address specific characteristics of named platforms, ports, or other entities. Implicit queries, on the other hand, do not name the referent entity but instead stipulate a condition that the entity must satisfy.

Implicit. At least one of the referent entities is not named but only implicitly indicated by a stipulated condition. The following are acceptable examples:

What is the position of the closest U.S. ship to Honolulu?
What is the readiness and reason of all ships within 700 miles of PECOS?
List all merchants carrying vanadium ore.

Situation Display. All queries and commands (explicit or implicit) that access the Situation Display are deemed members of this class. The following are acceptable examples:

Select a map of 700 miles from PECOS.
Show all U.S. ships.
Erase the RATHBURNE.

Define Word. A feature of LADDER is its facility to allow the user to define his own terms, and such definitions give rise to two additional query types. In the Define Word command, LADDER is instructed to substitute a new word (or string of symbols) for an old word already in its vocabulary and to regard these as equivalent in all future queries. The following are acceptable examples:

Define JFK to be like KENNEDY.
Define HON to be like Honolulu.
Define P to be like PECOS.

Define Phrase. This construction instructs LADDER to substitute a new word (or string of symbols) for an old phrase already in its vocabulary. The definition must be made in the context of a query. LADDER answers the query and treats the word as equivalent to the phrase in future queries. The following are acceptable examples:

Define (range from KENNEDY to Honolulu)
. . . like (what is the distance from KENNEDY to Honolulu).

Define (list 700P)
. . . like (list all ships within 700 miles of PECOS).

Query Rejection as Related to Query Type

Figure 4 summarizes the data on query frequency and query rejection as these relate to the six types of queries. Thus, the Explicit--General Information queries were the most commonly used (33.6% of all queries), and Implicit queries were next most frequent (25.4%). All query types had a substantial rejection rate, but three were particularly error-prone: the Explicit--Time/Distance queries (46.2% rejection rate), the Define Word construction (42.9% rejection rate), and the Define Phrase query (42.9% rejection rate). Note that the Explicit--General Information queries (presumably the most straightforward) were just as likely to be rejected as the more "subtle" Implicit and Situation Display queries. Their respective rejection rates were 22.8, 23.7, and 26.0 percent.

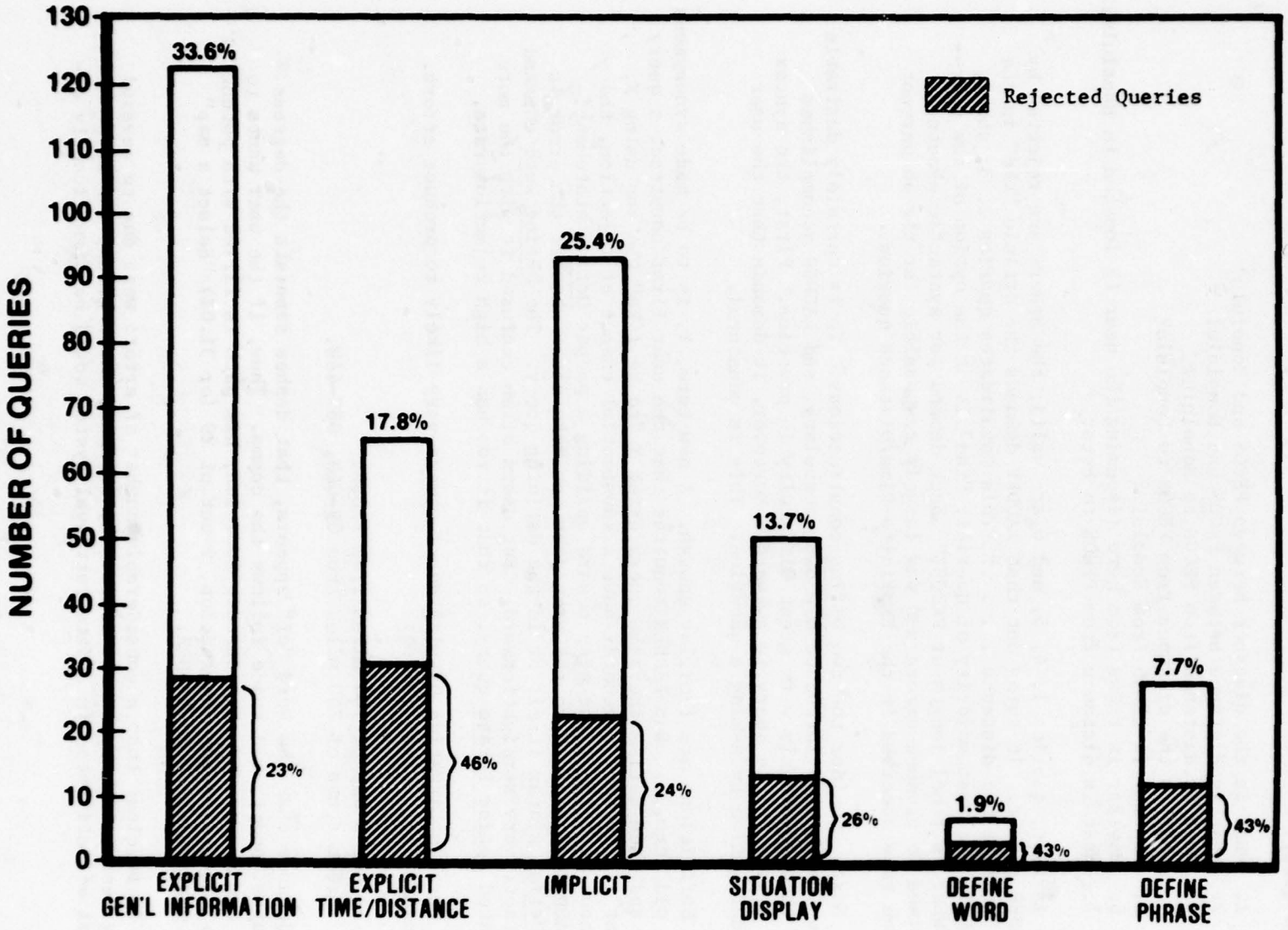


Figure 4. Query frequency and rejection rate.

It is most instructive to pinpoint some sources of query rejection. Consider first the following "natural" variations on an essentially simple distance query:

1. What is the distance between PECOS and Honolulu?
2. What is distance between PECOS and Honolulu?
3. What is distance from PECOS to Honolulu?
4. What is the distance from PECOS to Honolulu?
5. How far is PECOS from Honolulu?
6. How far is PECOS from here (assuming the user is located in Honolulu)?
7. What is distance from PECOS to here?

In fact, queries 1, 4, 5, and 6 are valid; the others are rejected by the LADDER parser. It turns out that LADDER demands the article "the" in the phrase "What is the distance . . . ." This invalidates queries 2, 3, and 7, although in the vast majority of queries, "the" is at the option of the user--e.g., "What is [the] length of PECOS?" Such demand for syntactic exactness is destined to produce errors and was largely responsible for the 46 percent rejection rate observed in the Explicit--Time/Distance queries.

Next, consider the two Define constructions. It is certainly desirable to allow the user to introduce his own vocabulary, and LADDER accomplishes this in theory but only with great difficulty in practice. First, the syntax for the Define Phrase query is awkward. Moreover, it demands that the user make a definition by asking a question. This is unnatural.

Definitions are familiar enough. A new term, Y, is to be made synonymous with an old term, X. But LADDER requires that the user first construct a query using X and then define the same query using Y "to be like" the one using X. Note that if queries in general have a substantial chance of rejection, the user cannot take the first step toward building a proper Define statement! Also, LADDER's feedback to the user does not make clear whether the error is in the Define syntax itself or in the defining query. The Define Word command is, in fact, very straightforward. But users often confused it with the more complicated Define Phrase query, so that it too had a high rejection rate.

Certain Situation Display queries were very likely to produce errors. Two illustrative queries follow:

Select a map of 700 miles from PECOS.
Select a map of 700 miles from 09-06N, 087-44W.

LADDER demands that the word "of" appears, that dashes separate the degrees and minutes, and that a space follows the comma. Thus, if the user wants to produce a map centered at a certain location, his path is filled with potential hazards. In the present evaluation, 9 out of 29 (or 31.0%) "Select a map" queries were rejected by LADDER.

It is clear that a considerable number of errors were due to several severely restrictive syntactic rules. By relaxing these rules, LADDER's potential effectiveness in a data retrieval system would be significantly enhanced.

Component Times for User-LADDER Interactions

It should be clear that the user and LADDER together comprise a man-machine query system. In this section the times required for the several component processes of this system are analyzed with an eye toward identifying those areas in which improvements might be made. It is assumed that requests for information originate outside of the query system proper, presumably with some external decision maker. The processes internal to the system consist of query formulation, query entry, parsing of the query, retrieval and presentation of the data, and the transfer of the accessed data back to the originator of the request. The response times for the first four of these operations are examined here.

Query Formulation

This measure was defined as the elapsed time between the display of the information request and the entry of the first character of the query. Further formulation of the query might certainly take place after the user starts typing, but no measure of such activity could be obtained.

Data were available for 304 of the 366 total queries.[2] The overall mean time for query formulation was 14.7 seconds, with a standard deviation of 19.4 seconds. Data for the individual users appear in columns 2, 3, and 4 of Table 2. The mean query formulation times for the users ranged from 7.5 to 32.1 seconds.

During the scenario, the users were provided with a reference folder containing much of the LADDER grammar tutorial, and they were encouraged to consult it as necessary. The folder was reported to be used in composing only 9.0 percent of the queries.

Query Entry

This measure was defined as the elapsed time between the first and last keystrike. It depends, of course, on the user's typing ability and his familiarity with similar keyboard entry systems. As mentioned above, final formulation of the query might also occur during this period, but such activity was always included in the entry time. Data were available for 315 of the 366 total queries. The overall mean time for query entry was 32.2 seconds; the standard deviation was 27.4. Data for the individual users appear in columns 5, 6, and 7 of Table 2. The mean entry times for the sample of users ranged from 18.6 to 48.0 seconds.

Query Parsing

This measurement identified the time from which the query was routed to LADDER to the receipt of the "PARSED" message. For the 258 parsed queries, the overall mean parsing time was 14.3 seconds, with a standard deviation of 13.8 seconds. Data for the individual users appear in columns 2, 3, and 4 of Table 3.

---

[2]The remaining data were lost due to magnetic tape storage errors.

Table 2

Query Formulation and Entry Times

	Query Formulation			Query Entry		
User #	Mean Time (secs)	SD	N	Mean Time (secs)	SD	N
1	10.8	20.8	38	34.9	27.0	38
2	32.1	33.9	9	33.9	24.0	10
3	16.2	12.6	26	41.5	31.4	26
4	7.5	10.7	28	21.9	16.1	28
5	20.1	24.0	31	27.3	24.8	32
6	7.8	7.4	42	18.6	13.8	46
7	12.6	10.4	31	39.7	31.8	32
8	17.9	19.2	37	34.6	23.9	39
9	21.6	18.6	27	48.0	40.8	28
10	15.9	26.1	35	30.1	21.0	36
All Users	14.7	19.4		32.2	27.4	

Table 3

LADDER's Parse and Retrieval Times

	Query Parsing			Data Retrieval		
User #	Mean Time (secs)	SD	N	Mean Time (secs)	SD	N
1	14.0	13.1	18	50.5	29.7	17
2	10.8	7.7	19	47.5	46.4	19
3	14.9	10.8	24	48.1	34.0	24
4	14.5	12.9	28	38.1	23.9	25
5	13.4	9.9	28	46.2	23.2	28
6	15.8	17.7	49	37.6	21.2	49
7	12.3	9.4	23	42.4	22.6	23
8	20.4	19.3	21	48.9	24.8	21
9	15.5	16.7	25	39.8	21.9	25
10	9.8	6.9	23	31.6	13.6	23
All Users	14.3	13.8		42.2	26.9	

Data Retrieval

This measure denoted the time elapsed between the "PARSED" message and the receipt of an answer from the host computer. Therefore, it included the access of the data base. For all users combined, the mean retrieval time was 42.2 seconds; the standard deviation was 26.9 seconds. Data for the individual users appear in columns 5, 6, and 7 of Table 3.

Data for the component times are summarized in the left-hand panel of Figure 5. Thus, the total time required for a typical successful interaction was approximately 105 seconds. The typical component times were approximately 15 seconds for query formulation, 30 seconds for query entry, 15 seconds for parsing of the query, and 45 seconds for retrieval of the answer.

The overall mean transaction time of 1.75 minutes for successful queries seems quite impressive when one considers the complexity and amount of data being accessed by the system. Such data retrieval would certainly require much more time using current Navy procedures.

Query Rejection

Since the queries that LADDER failed to parse comprised 29.5 percent of the query population, a vital datum is the time required for LADDER to reject a query as illegal. For 106 of the 108 rejected queries[3] the mean time to reject was 38.4 seconds; the standard deviation was 44.7 seconds. The right-hand panel of Figure 5 depicts the typical cycle for rejected queries (assuming that entry and formulation times for the rejected queries are essentially those that hold for the total query population).

Notice that, on the average, LADDER takes more than 2.5 times as long to reject a query as faulty (38.4 seconds) than it does to parse a legal entry (14.3 seconds). Overall, the time-to-reject is fully two-thirds that required for parsing and retrieval (56.5 seconds). It follows that LADDER's rejection algorithm is a substantial source of overall transaction time. And of course, each rejected query must be reformulated and reentered, so that the price paid for query rejection is indeed high.

Truncation of the Parsing Algorithm

An obvious remedy to the high cost of rejections is to reduce the number of illegal queries. One approach is to modify the parser so that it would admit those queries that are essentially "good" but now violate LADDER's syntactic rules. This would no doubt entail substantial reprogramming of the parser.

---

[3]The two omitted queries, in fact, resulted in system failures.

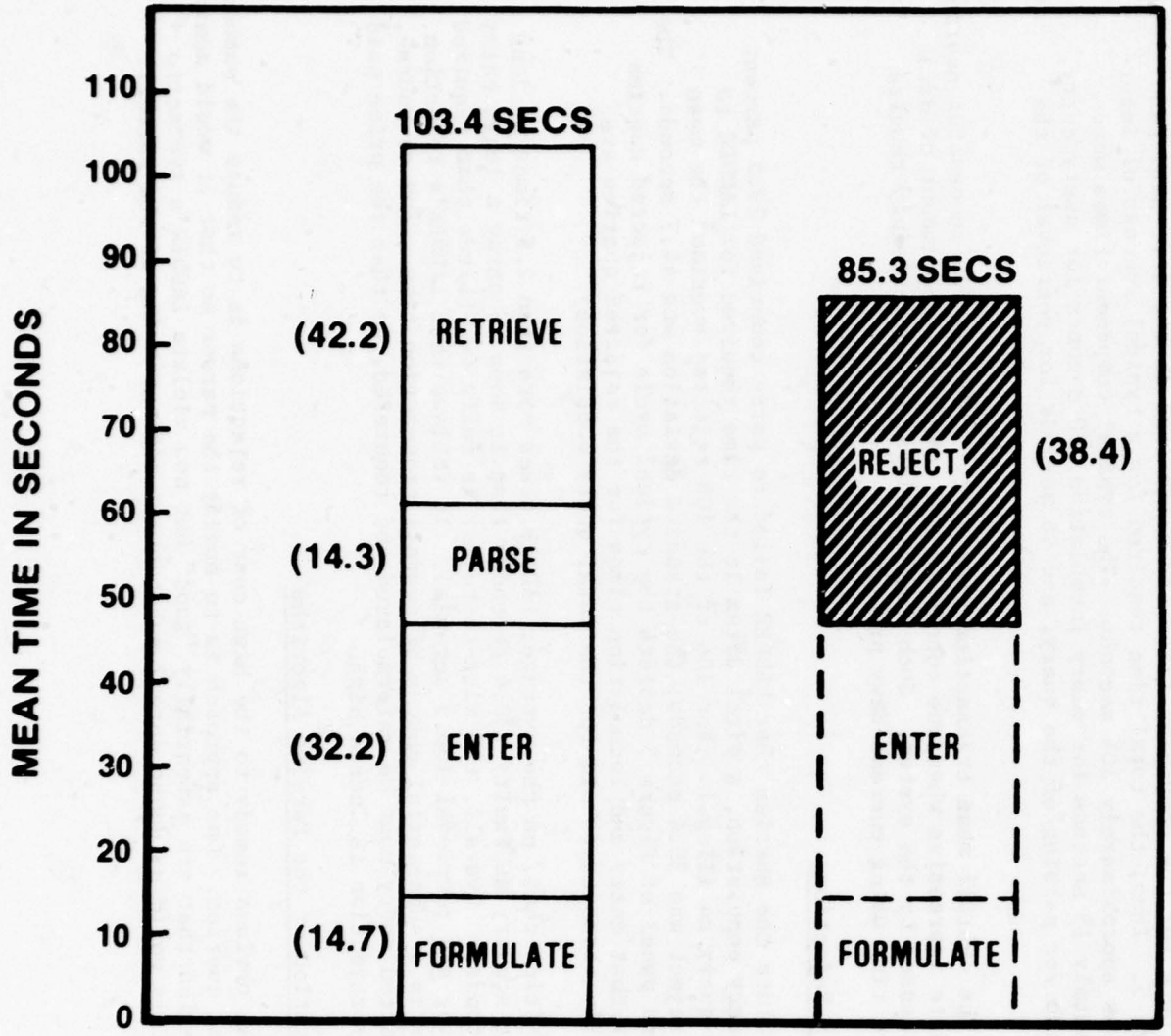


Figure 5. Component times for user-LADDER interactions.

A more straightforward option is to reduce LADDER's rejection time by simply exiting the parsing routine after x seconds, if LADDER has failed to either parse or reject by that time. Note that LADDER's current mode of operation is to exit only on a successful parse or on a definite rejection. The result is that the parser will dwell excessively on a given query. For instance, six queries (one of which was parsed) were processed for more than 100 seconds. At the extreme, one query was finally rejected after 5.1 minutes of processing.

Putting a limit on the maximum time allowed for parsing or rejecting a query will, in general, reduce the mean rejection time. However, LADDER may then fail to parse some legal queries (those that would be parsed in $> x$ seconds). The details of this tradeoff can be explored for the current data by using Figure 6, which gives the cumulative probability distributions (as a percentage) for both parsed and rejected queries.

For example, consider terminating the parsing algorithm at $x = 120$ seconds. Since all successful queries were parsed in less than 120 seconds, these queries would be unaffected. For the rejected queries, 4.7 percent (five queries) had rejection times exceeding the cutoff. Taking into account the actual rejection times for such queries and assigning them the maximum value of 120 seconds, the mean rejection time would be reduced from 38.4 to 35.0 seconds. This small savings amounts to an 8.8 percent reduction. Similar effects were computed for various values of the cutoff x and appear in Table 4.

Notice, for instance, that with $x = 80$ seconds, less than 1 percent of the parsed queries would be lost, and the mean rejection time would be reduced from 38.4 to 32.1 seconds, a savings of 16.4 percent. Even with $x = 20$ seconds, 84.8 percent of the parsed queries would still be retained, and the mean rejection time would be lowered by 53 percent. No specific cutoff is recommended, but subsequent revisions to LADDER might well be based on these tradeoffs.

A plausible tactic might be to truncate the parsing algorithm but also give the user the option of reentering the query--if he detects an obvious error or can reformulate the query in more simple syntax--or allowing the algorithm to proceed as before. Such truncated queries would suffer somewhat from the lack of diagnostic feedback that LADDER usually provides for rejected queries. However, the present evaluation suggests that LADDER's diagnostic feedback is often cryptic and is not a powerful aid to the user.

Error Recovery and Inference by LADDER

LADDER has the capability to correct certain spelling errors made by the user and to make inferences for selected queries that it deems ambiguous or incomplete. The users made a total of 16 spelling errors (4.4% of all queries); ten of these were properly corrected by LADDER.[4] For example, LADDER was able to correctly recognize "wner" as "owner" and "RATHBONE" as "RATHBURNE." However, on two occasions, LADDER "corrected" words that were already spelled properly, and these resulted in rejected queries. Thus, LADDER "corrected" the word "all" to "call" on one occasion and to "LA" on another.

---

[4]Spelling errors were not counted as such if the query had been rejected before the parser reached the given error.

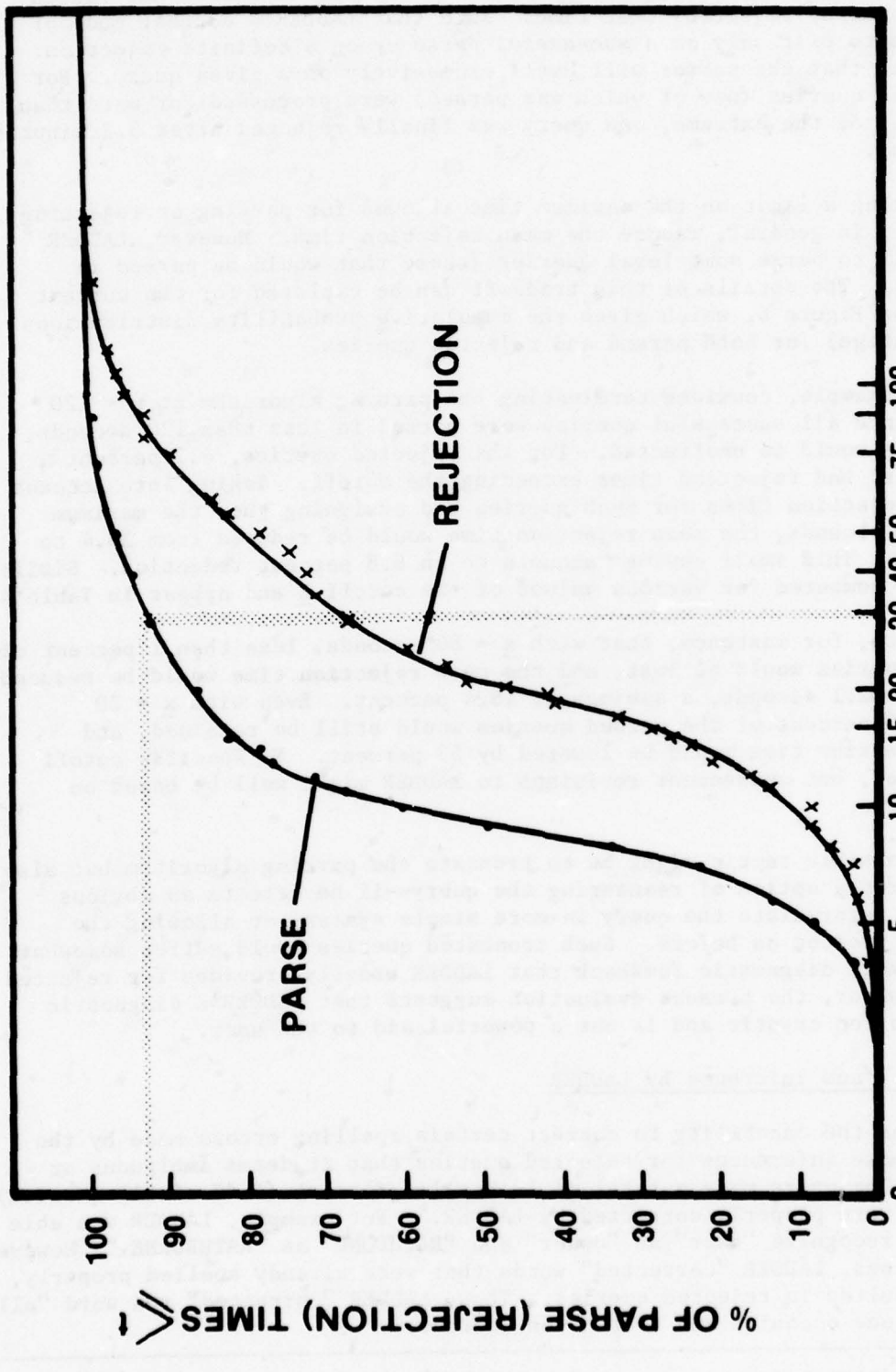


Figure 6. Cumulative distribution functions for parse and rejection times.

Table 4

The Effects of Imposing a Time Limit on the Parsing Algorithm

Value of the Cutoff (secs)	% of Parsed Queries Retained	% of Rejected Queries < Cutoff	Mean Rejection Time (secs)	% Savings in Rejection Time
120	100.0	95.3	35.0	8.8
100	99.6	92.5	33.7	12.2
90	99.6	91.5	33.0	14.1
80	99.2	89.6	32.1	16.4
70	98.8	86.8	30.9	19.5
60	97.3	84.0	29.5	23.2
50	96.5	78.3	27.7	27.9
40	94.2	71.7	25.2	34.4
30	91.9	66.0	22.2	42.2
20	84.8	41.5	18.0	53.1

LADDER never recovered if the user omitted a space (as in "700miles"); the eight omissions of a space were not counted as errors in spelling.

LADDER inferences are of two types: (1) those that refer to a previous query, as in "What is the length of the KENNEDY?" followed by "Her home port?"; and (2) those that implicitly refer to the fixed location of the user, as in "How far is PECOS?"--which LADDER takes to mean "How far is PECOS from Honolulu?" In all, 54 (or 14.8%) of the users' queries resulted in inferences by LADDER; 41 of these were of type (1) above and 13 were of type (2). LADDER was able to recover--that is, generate a parseable query--for all of the type (1) inferences and for 46.5 percent of the type (2) queries. The latter often had other syntactic problems or induced an inference by LADDER that was clearly not intended by the user.

Queries Resulting in System Failure

It is, of course, desirable that for any given query, LADDER either process it to a successful retrieval or reject it as illegal. However, the system should continue to run in either case. In the present evaluation or in preliminary tests, seven different queries had the effect of "crashing" the system. These "catastrophic" queries are given below with the intention that any future revisions to LADDER might make the system less vulnerable.

1. List distances from tg to PECOS.
   (Note: "tg" had been defined as "all ships within 700 miles of PECOS.")

2. List all U.S. ships.

3. Define (where is sar-1)
   . . . like (where is RATHBURNE and KNOX).

4. Define (where is sar-1)
   . . . like (where is RATHBURNE, KNOX).

5. What is the distance from PECOS to CONSTELLATION, BIDDLE, RK TURNER, HALSEY, REEVES, WILSON, KNOX, RATHBURNE, HASSAYAMPA, BRITISH CAPTAIN, ADELAIDE STAR?

6. What is opcon?

7. Where is the opcon of sar-1?

It would appear that queries 2, 3, and 4 are legal. Thus, their failure might be simply coincident with a failure of the operating system, or LADDER might somehow be sensitive to its past history (i.e., those queries that immediately precede the catastrophic failure.

Users' Evaluations of LADDER

The users' opinions about LADDER were solicited by questionnaire at the conclusion of each test run. A copy of this questionnaire is provided in Appendix B along with the users' median response to each of the scaled items. The results are summarized below. In general, they reinforce the objective measures of performance already discussed.

LADDER Training Session

Despite its relatively short duration, the users reported that the training session provided an adequate introduction to LADDER's grammar, vocabulary, and editing procedures. They believed that the length of the tutorial was appropriate but that additional practice with LADDER's grammar would have been desirable. In particular, many of the users indicated that special attention should have been directed toward understanding LADDER's syntactic idiosyncrasies. The median estimate of the time required to become proficient in LADDER was 10.5 hours, with estimates ranging from 1 hour to 1 week.

LADDER Syntax

Generally, the users were favorably impressed by LADDER's features and its ability to understand normal, conversational English. However, limitations in LADDER's sensitivity to command control vocabulary and abbreviations were reported to be incompatible with typical command center operating procedures. The users indicated that the Define Word construction was the least useful of LADDER's special features. The ability to ask compound queries, to refer to prior queries, and to define phrases were all considered to be quite desirable. The users reported that LADDER's syntactic features were fairly easy to use, except for the Time/Distance and Define queries. The difficulties surrounding the use of these query types were attributed to their restrictive and esoteric syntactic rules.

LADDER Output

All users reported that the time required by LADDER to retrieve the requested data or to reject a query was excessive. The format of LADDER's answers was judged to be satisfactory, but the diagnostic information that accompanies a rejected query was considered to be of little use. The primary reason for this is that the diagnostic feedback was too often expressed in terms that were unfamiliar to the user.

Anticipated Operational Usage

The users indicated that LADDER would be very useful in an actual search and rescue operation, or in a wide variety of surveillance and planning functions. LADDER's speed in retrieving data was considered much faster than that of current procedures. Thus, although the users found LADDER's response time excessive, they recognized that it was quite efficient in comparison to other systems.

The users reported that LADDER would be employed primarily by trained operators acting in support of a decision maker. The decision maker might use the system occasionally during inactive periods, but would rely routinely on the operators to retrieve needed data. It was suggested that the decision maker's need to consider all aspects of the operational situation would preclude his having substantial involvement with LADDER.

CONCLUSIONS AND RECOMMENDATIONS

The LADDER prototype system demonstrated impressive capabilities in interpreting natural language and retrieving information from a command control data base. However, at its current stage of development, LADDER's natural language subset is less than completely "natural," as evidenced by its 29.5 percent query rejection rate.

Selected types of queries prove very prone to error because of LADDER's rigid syntactical demands. The queries involving time or distance could be improved by enlarging the permissible grammatical patterns. The user functions for defining words and phrases should be made more flexible and simpler to use. Such syntactical problems could no doubt be resolved by more intensive training of would-be users, but this is deemed contrary to the goals of an advanced natural language query system.

LADDER's lexicon does not include such familiar naval terms as "range," "nm," and "dst." Minor revisions to LADDER should accomodate these usages.

LADDER's rejection algorithm should be improved in order to reduce the excessive time that is required to determine that a query is faulty. It is suggested that the algorithm exit after a predetermined time if the query has not yet been parsed. Appropriate tradeoffs between savings in time and the rejection of valid queries can be explored based on the data developed here.

Fleet applications of natural language query systems must await the evolution and refinement of the LADDER prototype technology. Objective evaluations of system performance (in contrast to "demonstrations" and "subjective assessments") can best contribute to this evolution. The performance of LADDER in other command control scenarios should be evaluated.

REFERENCES

Navy WWMCCS software standardization (NWSS) (Vol. II, Book 3, Query module users manual, Doc. No. 07A 001A; UM-04, draft). Washington, DC: Naval Command Systems Support Activity, 1975.

Sacerdoti, E. D. (Ed.). Mechanical intelligence: Research and applications. Menlo Park, CA: SRI International, December 1977.

APPENDIX A

INFORMATION REQUESTS TO THE OPERATOR

INFORMATION REQUESTS TO THE OPERATOR

INFORMATION REQUEST 1

Find the following operational information on the PECOS . . .
What is her nationality?
Then find out her owner.

INFORMATION REQUEST 2

We need an area search for the nearest ships to the PECOS.
Plot all ships within 200 miles.
Her last position was 73-40N, 174-30W.
Also find out how far the PECOS is from here.

INFORMATION REQUEST 3

Since there are no nearby ships, you had better expand that search and get a listing of those ships within 700 miles of the PECOS.

INFORMATION REQUEST 4

These are our candidate SAR ships . . .
Find the employment schedules and fuel status of the ships on that list.

INFORMATION REQUEST 5

Determine their distances from the PECOS.

INFORMATION REQUEST 6

Determine the cruising endurance and maximum range of each of these ships.

INFORMATION REQUEST 7

I want to know some operational information about those candidate ships to help me in making a task assignment.
Find their readiness and reason.

INFORMATION REQUEST 8

It appears that there are some readiness problems with those ships.
Check the data base for readiness, reason, casrep, and eicnoms.
Get this in one list on your display . . .

INFORMATION REQUEST 9

Which of these ships has a doctor aboard?

INFORMATION REQUEST 10

I need some more information on the PECOS.
Determine her ship class, gross weight, and dead weight.

INFORMATION REQUEST 11

Also find her port of departure and then her destination port.

INFORMATION REQUEST 12

My first choice for the ships for this SAR is the RATHBURNE backed up by the KNOX because of the readiness status and proximity.
Tell me how long it will take for those ships to reach the PECOS.

INFORMATION 13

I want you to designate the RATHBURNE and the KNOX as SAR-1.
Find the OPCONS of SAR-1.

INFORMATION REQUEST 14

Find the names of the commanding officers of SAR-1 . . .

INFORMATION REQUEST 15

Here is a summary of the situation . . .

I have requested the S3-A from the CONSTELLATION to proceed to the PECOS and report the situation. The RATHBURNE and KNOX have been tasked and are proceeding to the area. The units got underway within 20 minutes after receipt of their orders. We should expect reports from these units shortly . . . In the meantime, please determine the radio call signs of SAR-1.

CONCLUDING MESSAGE:

The following message was received from the RATHBURNE:

SAR SITREP ONE

1. DIRECTED TO SCENE BY SCHOOLBOY 63 WHO ESTABLISHED VISUAL CONTACT WITH THE PECOS AT 011714Z5. AIRCRAFT REPORTED SHIP BURNING FROM STERN HOLD BUT SHIP STABLE, NO CREWMEN IN RAFTS; THOSE NOT FIGHTING FIRE LOCATED ON BOW.

2. MADE VISUAL CONTACT AND ASSUMED ON-SCENE-CDR AT 811722Z3 COMMENCED ASSISTING IN FIGHTING FIRE AT 011810Z1 AT 37-35NB, 174-37WG. KNOX ARRIVED ON SCENE AT 011755Z9.

3. FIRE UNDER CONTROL AT 011821Z3. FOURTEEN PECOS CREWMEN TRANSFERRED TO KNOX VIA HELO FOR TREATMENT OF MINOR AND SERIOUS BURNS BY KNOX DOCTOR AND HMC. NO REPEAT NO CASUALTIES.

4. PECOS MASTER REPORTS UNABLE TO STEAM UNDER OWN POWER, AND HAS CONTACTED OWNER (TEXACO) WHO IS DISPATCHING TUG FROM HONO TO TOW PECOS INTO PORT.

5. UNODIR DETACHING KNOX TO PROCEED BEST SPEED TO MIDWAY WITH INJURED CREWMEN AND WILL REMAIN WITH PECOS TO ESCORT HER AND TUG INTO PORT.
BT

APPENDIX B

DEBRIEFING QUESTIONNAIRE

DEBRIEFING QUESTIONNAIRE

Your reactions to LADDER are an important part of our evaluation. Please take your time and complete this questionnaire. Most items ask you to circle that number (1, 2, 3, 4, or 5) which best describes your opinion. However, feel free to write any additional remarks.

1. After the training session but before the scenario, did you feel prepared to use

	unprepared				very well prepared	
LADDER grammar	1	2	3	4	5	(3.5)[1]
LADDER vocabulary	1	2	3	4	5	(3.5)
LADDER editing	1	2	3	4	5	(4.0)
general LADDER procedures	1	2	3	4	5	(4.0)

2. Now that you've completed the scenario, how did LADDER compare with your expectations based on the training session?

Easier than expected		What I expected		Harder than expected	
1	2	3	4	5	(3.5)

3. How long do you think it would take someone like yourself to become proficient (not perfect) with LADDER? (insert numbers)

_____ minutes _____ hours _____ days _____ weeks _____ months (10.5 hrs.)

4. The length of the training session was

too short		about right		too long	
1	2	3	4	5	(3.0)

5. In this scenario, how desirable was it to

	not at all				very desirable	
ask compound queries	1	2	3	4	5	(4.0)
refer back to earlier queries	1	2	3	4	5	(4.0)
redefine words, i.e., "define xyz to be like abc"	1	2	3	4	5	(2.0)
construct your own language, i.e., "define (xyz) like (abc)"	1	2	3	4	5	(4.0)

[1]Numbers in parentheses indicate median response.

6. In terms of their ease of use, how would you rate

	easy to use				difficult to use	
normal status information queries	1	2	3	4	5	(1.0)
compound queries	1	2	3	4	5	(2.0)
queries which referred to previous ones	1	2	3	4	5	(2.5)
time - distance queries	1	2	3	4	5	(4.0)
situation display queries	1	2	3	4	5	(2.0)
"define xyz to be like abc"	1	2	3	4	5	(3.0)
"define (xyz) like (abc)"	1	2	3	4	5	(3.0)

7. How well did you like LADDER's ability to understand normal, conversational English?

not at all				very much	
1	2	3	4	5	(4.0)

8. In your opinion, how well would LADDER accommodate the type of language used in Navy command and control missions?

poorly				very well	
1	2	3	4	5	(3.5)

9. Rate the usefulness of LADDER's editing features.

	not useful				very useful	
rubout	1	2	3	4	5	(5.0)
kill query	1	2	3	4	5	(5.0)

10. Would you like to have additional editing capabilities? If yes, describe.

11. When LADDER is unable to answer a query, it gives some diagnostic information. How useful did you find this information in rewording your query?

not at all				very useful	
1	2	3	4	5	(2.5)

12. How well could you understand LADDER's answers?

not at all				very well	
1	2	3	4	5	(4.0)

13. What did you think of the format of LADDER's answers?

easy to read — hard to read

1 2 3 4 5 (2.0)

14. How did you feel about the time required by LADDER to answer your queries?

too long — about right — very fast

1 2 3 4 5 (2.0)

15. What recommendations, if any, would you make to improve LADDER? Note any difficulties you had, annoying limitations, missing features, etc.

16. How useful do you think LADDER would be in an actual SAROPS situation?

not at all — very useful

1 2 3 4 5 (4.5)

17. In what other situations, if any, do you feel that LADDER might be useful?

18. Is it likely that an officer decision-maker would <u>personally</u> use LADDER in a command and control situation? (Or would he rely on a specially trained LADDER operator?)

mainly used by operators — mainly used by decision-makers

1 2 3 4 5 (2.0)

Explain:

19. How does LADDER's speed in retrieving data compare with current procedures?

LADDER is much slower — LADDER is much faster

1 2 3 4 5 (4.5)

20. Your name:
Duty station:
Phone number:
Months/years of command and control experience:

Are you experienced in SAROPS? Yes No

How often do you use computers?

daily		every few months		almost never	
1	2	3	4	5	(3.5)

Do you know any computer programming languages? If yes, name them.

THANK YOU!

DISTRIBUTION LIST

Chief of Naval Operations (OP-102) (2), (OP-11), (OP-094), (OP-987H)
Chief of Naval Material (NMAT 08T244)
Chief of Naval Research (Code 200), (Code 230), (Code 431), (Code 434), (Code 436), (Code 437), (Code 450) (4), (Code 452), (Code 458) (2)
Chief of Information (OI-2252)
Director of Navy Laboratories
Commandant of the Marine Corps (Code MPI-20), (Code CC)
Chief of Naval Education and Training (N-5)
Chief of Naval Technical Training (Code 016)
Commander Training Command, U. S. Atlantic Fleet (Code N3A)
Commander, Naval Data Automation Command
Commander, Naval Military Personnel Command (NMPC-013C)
Commander, Naval Electronic Systems Command
Commander, Naval Ocean Systems Center (Code 83), (Code 91), (Code 447), (Code 823), (Code 8321)
Commanding Officer, Fleet Combat Training Center, Pacific
Commanding Officer, Fleet Combat Training Center, Pacific (Code 00E)
Commanding Officer, Naval Education and Training Program Development Center (2)
Commanding Officer, Naval Development and Training Center (Code 0120)
Commanding Officer, Naval Training Equipment Center (Technical Library)
Commanding Officer, Navy Regional Data Automation Center, Washington, Command and Control Systems Directorate
Commanding Officer, Office of Naval Research Branch Office, Boston
Commanding Officer, Office of Naval Research Branch Office, Chicago
Commanding Officer, Office of Naval Research Branch Office, Pasadena
Director, Training Analysis and Evaluation Group (TAEG)
Superintendent, Naval Academy
Superintendent, Naval Postgraduate School
Personnel Research Division, Air Force Human Resources Laboratory, Brooks Air Force Base
Technical Library, Air Force Human Resources Laboratory, Brooks Air Force Base
Advanced Systems Division, Air Force Human Resources Laboratory, Wright-Patterson Air Force Base
Human Engineering Division, Wright-Patterson Air Force Base
Program Manager, Life Sciences Directorate, Air Force Office of Scientific Research
Assistant Secretary of the Air Force (Research, Development, and Logistics)
Deputy Chief of Staff for Research, Development, and Acquisition, Department of the Air Force
Command Control and Communications, Air Force Test and Evaluation Center, Kirtland Air Force Base
Deputy for Communications and Information Systems, Electronic Systems Division, Hanscom Air Force Base
Chief, Information Sciences, Rome Air Development Center, Griffiss Air Force Base
Commander, 6570th Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base
Assistant Secretary of the Army (Research, Development, and Acquisition)
Assistant Chief of Staff for Automation and Communications, Department of the Army

Army Research Institute for the Behavioral and Social Sciences
Army Research Institute for the Behavioral and Social Sciences
Field Unit--USAREUR (Library)
Communication and Electronic Readiness Command, Department of the Army,
Fort Monmouth
Director, Command and Control Technical Center
Deputy Director for Operations (Command, Control, and Communications),
Joint Staff, Organization of the Joint Chiefs of Staff
Assistant Secretary of Defense (Communications, Command, Control, and
Intelligence)
Director, Environmental and Life Sciences, Office of the Under Secretary
of Defense for Research and Engineering
Director, Information Processing Techniques Office, Defense Advanced
Research Projects Agency
Director, Cybernetics Technology Office, Defense Advanced Research
Projects Agency
Military Assistant for Training and Personnel Technology, Office of
the Under Secretary of Defense for Research and Engineering
Commandant, Industrial College of the Armed Forces
Coast Guard Headquarters (G-P-1/62)
Defense Documentation Center (12)